Cognitive GPR for Subsurface Object Detection Based on Deep Reinforcement Learning

Maxwell M. Omwenga, *Graduate Student Member, IEEE*, Dalei Wu, *Member, IEEE*, Yu Liang, Li Yang, Dryver Huston, and Tian Xia, *Senior Member, IEEE*

***Abstract*—Ground penetrating radars (GPRs) carried by mobile platforms, such as vehicles and drones, have been applied in various applications, for instance, subsurface utility detection, structural health inspection, and autonomous driving. However, existing GPR systems are not able to operate autonomously and adaptively due to several challenges, including the lack of intelligence, uncertain and dynamic nature of sensing environments, and huge state and action spaces. To overcome these challenges, in this article, we propose an autonomous cognitive GPR (AC-GPR) enabled by a deep reinforcement learning (DRL) approach. Specifically, the operation of the proposed AC-GPR is first formulated as a sequential decision process. A novel reward function is developed for the DRL model by defining and combining two different types of entropy-based rewards resulting from object detection and recognition, respectively. A deep Q-learning network (DQN) is developed to address the extreme curse of dimensionality in the state space and learn a policy directing the actions of the AC-GPR. The AC-GPR is evaluated using software called GprMax by combining DRL with GPR modeling and simulation. Results show that our proposed DRL-based AC-GPR outperforms other GPR systems using different approaches in terms of detection accuracy and operating time.**

***Index Terms*—Autonomous cognitive ground penetrating radar (GPR), deep reinforcement learning (DRL), subsurface sensing.**

I. Introduction

GROUND penetrating radars (GPRs) have been extensively used in many industrial applications, such as coal mining, structural health monitoring, subsurface utilities detection and localization, and autonomous driving [1], [2]. In subsurface detection applications, a GPR system transmits an electromagnetic (EM) wave into the ground at several spatial positions and receives the reflected signal to form GPR data, called A-Scans, B-Scans, and C-Scans with a different number of dimensions [1], [3]. As shown in Fig. 1, a single radar trace, or waveform, is called A-Scan, which is 1-D signal. A set of consecutive radar waveforms along a particular direction (for example, x-axis in Fig. 1) can be assembled into a 2-D image called B-Scan. Multiple B-Scan images along a particular direction (for example, y-axis in Fig. 1) can be composed into a C-Scan. In this article, we consider B-Scan because it is the most commonly used GPR data modality for subsurface object detection.

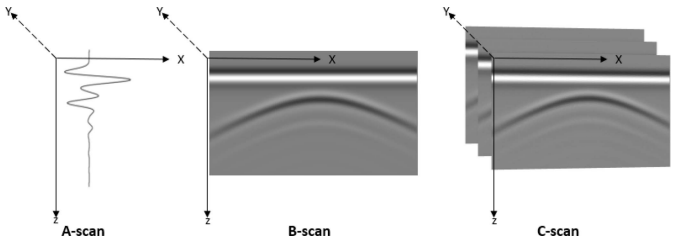

Fig. 1. 1-D A-Scan, 2-D B-Scan, and 3-D C-Scan.

Although GPRs are effective in many nondestructive applications, most of the existing GPR systems are human-operated due to the need for experience in operation configurations based on the interpretation of collected GPR data. GPR-based subsurface survey is complicated as various sensing environment and subsurface objects have dissimilar features. In actual GPR survey, GPR sensing quality could be affected by many factors, including environmental factors, such as soil dielectric properties, environment noise, clutter, multipath effects, combined near and far-field effects, and GPR operational system parameters, such as wavelength (or frequency), waveform, polarization, wave timing, transmitter and receiving antennas direct coupling, etc.

In addition, the subsurface objects have different structural features and EM properties that affect GPR EM wave propagation differently. Hence, processing GPR data and extracting information of interest are challenging and involve a series of sophisticated steps. In nearly all existing GPR systems, GPR data processing is performed offline where data from the field are collected and stored, and then postprocessed on a computer after the scanning. Such a processing approach is time consuming and lacks adaptivity. Also, some applications involve sensing tasks within hazardous and inaccessible environments.

To achieve optimal sensing performance, it is desired to design an autonomous GPR system that can operate adaptively under varying sensing conditions. Specifically, the system is able to adaptively move with a robotic platform and adjust its operational parameters through real-time interaction with

Manuscript received November 10, 2020; revised December 31, 2020; accepted January 27, 2021. Date of publication February 15, 2021; date of current version July 7, 2021. This work was supported by the National Science Foundation under Grant 1647175 and Grant 1924278. *(Corresponding author: Dalei Wu.)*

Maxwell M. Omwenga, Dalei Wu, Yu Liang, and Li Yang are with the Department of Computer Science and Engineering, University of Tennessee at Chattanooga, Chattanooga, TN 37403 USA (e-mail: dgh179@mocs.utc.edu; dalei-wu@utc.edu; yu-liang@utc.edu; li-yang@utc.edu).

Dryver Huston is with the Department of Mechanical Engineering, University of Vermont, Burlington, VT 05405 USA (e-mail: dryver.huston@uvm.edu).

Tian Xia is with the Department of Electrical and Biomedical Engineering, University of Vermont, Burlington, VT 05405 USA (e-mail: txia@uvm.edu).

Digital Object Identifier 10.1109/JIOT.2021.3059281

2327-4662 © 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: UTC Library. Downloaded on October 01,2021 at 17:00:56 UTC from IEEE Xplore. Restrictions apply.

the sensing environment. Although the decision-making process of operating an autonomous GPR can be modeled as a finite-horizon Markov decision process (MDP) with finite state and action spaces, the curse of extremely high dimensionality of state space makes it computationally infeasible to derive optimal action using the standard infinite-horizon MDP algorithms [4]. Deep reinforcement learning (DRL) is suitable for solving this problem since it can reduce the dimensionality of the large state space while learning the optimal policy at the same time.

This article is focused on the development of a DRL framework that enables AC-GPR. To this end, a proper reward function is needed to effectively reflect the value of different actions of the AC-GPR agent at different states. Also, a DRL algorithm with the reward function needs to be developed to learn a policy that directs the AC-GPR's actions. To the best of our knowledge, this is the first work based on DRL for the development of AC-GPR. The main contribution of this article can be summarized as follows.

1) By formulating cognitive subsurface object detection as a Markov decision problem, a DRL framework is established to resolve the problem.
2) A deep Q-network (DQN) algorithm with a novel reward function that combines rewards from both region of interest (RoI) identification and object classification is proposed.
3) To show the efficacy of the proposed framework, simulation-based validations are performed on real-time GPR data from GprMax simulator by combining DRL with GPR operation modeling [5], [6].

The remainder of this article is organized as follows. The related work is presented in Section II. In Section III, an overview of the system model and architecture of the proposed AC-GPR is presented. The proposed DRL approach is discussed in Section IV. Section V presents performance evaluation and discussion. Finally, Section VI concludes this article.

II. Related Work

The Internet of Things (IoT) and robotics fields are linking up to forge Internet of Robotic Things (IoRT) [7] where smart gadgets can screen the events occurring around them, intertwine their sensor information, utilize local and/or distributed intelligence to autonomously plan the course of action(s) to control gadgets in the physical world. For example, the complexity of autonomous driving on urban roadways is addressed by applying RL method [8] and DRL [9].

Autonomous GPRs have been extensively studied in field robotics, remote sensing, and intelligent transportation systems [10]–[15]. Cornick *et al.* [10] described a localizing GPR system fused with GPS, LiDAR, and camera hooked at the bottom of an autonomous vehicle for autonomous ground vehicle localization. The system allows real-time creation of single-track maps with online data processing, as well as real-time localization of the vehicle to a prior map. Williams *et al.* [11] developed an autonomous robotic system employing GPR probing of glacier surfaces for void detection in ice. Supervised machine learning with pretrained models was applied to automatically classify data into crevasse and crevasse-free classes. Other machine learning techniques have been applied to analyze B-Scans for object detection using Faster R-CNN [12], and incorporating frequency domain features in classification with augmented GPR data synthesized by the generative adversarial network (GAN) [13]. Foessel-Bunting *et al.* [14] described a sled-mounted GPR integrated with position and latitude instrument for autonomous search for Antarctic meteorites. Using nondestructive evaluation (NDE) sensors, histogram of gradient (HOG) and Naive Bayes classifier, Le *et al.* [15] developed an autonomous robotic system for real-time bridge deck inspection that generates condition map. Although the aforementioned systems have used robotic systems to move GPR scanners, GPR movement and its operational parameters were not adaptively adjusted on the fly.

DRL techniques in autonomous robots and sequential decision making process have been broadly studied [16]–[18]. The current studies incorporate the versatile operation control of robots, mechanical control, and the administration in multirobot frameworks locally, at the edge or the cloud. Xin *et al.* [19] used a DQN to develop an end-to-end autonomous robotic system that incorporates path planning. Robots and humans safely coexist through socially compliant interaction with inverse reinforcement learning (RL) [20]. The solution for mission-driven robotics with visual navigation problems has been developed through DRL and AI2-THOR framework in [21].

Recent research, such as automatic exploration for navigation in unknown environment using DRL-based decision algorithm with classical robotic methods [22] is gaining attention. Vehicle classification with the DRL algorithm that selects key areas from an image automatically, through integrating multiglimpse and visual attention mechanism which highlights one part of an image and weaken the others [23].

By formulating active object detection as a sequential action decision process, the work in [24] implemented a hybrid of deep Q-learning network (DQN) with a dueling. DRL was also studied in a sequential decision making process for imbalance classification [25] by formulating a sensitivity reward function for minority class data sets. An artificial agent is trained to act in a human-like manner to reduce over-segmentation errors through a joint surgical gesture segmentation and classification method [26].

III. Proposed System Model and Architecture

A. Original Concept of Cognitive GPR

A typical operational scenario is that a GPR moves around in a predetermined geographical area (environment) to detect a subsurface object through transmitting EM waves into the ground, and receiving reflected EM waves whenever there's a contrast in material dielectric properties, as shown in Fig. 2. Adaptive tuning of operational parameters of the AC-GPR, such as frequency, waveform, polarization and wave timing, may lead to considerable improvement in the quality and

Authorized licensed use limited to: UTC Library. Downloaded on October 01,2021 at 17:00:56 UTC from IEEE Xplore. Restrictions apply.

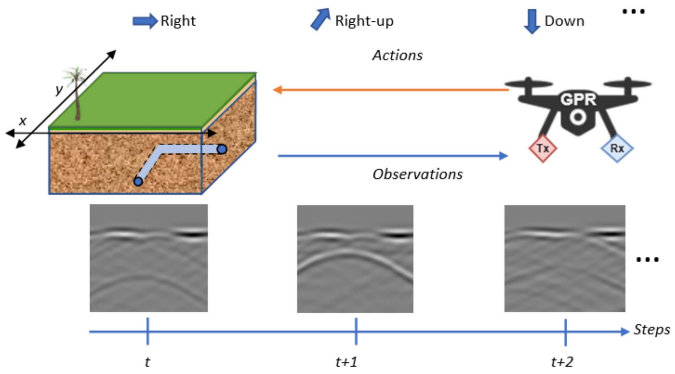

Fig. 2. Successive action choice procedure of dynamic subsurface object detection, where the GPR takes a perception under certain state, and executes an action, and receives an observed B-Scan image at time step t.

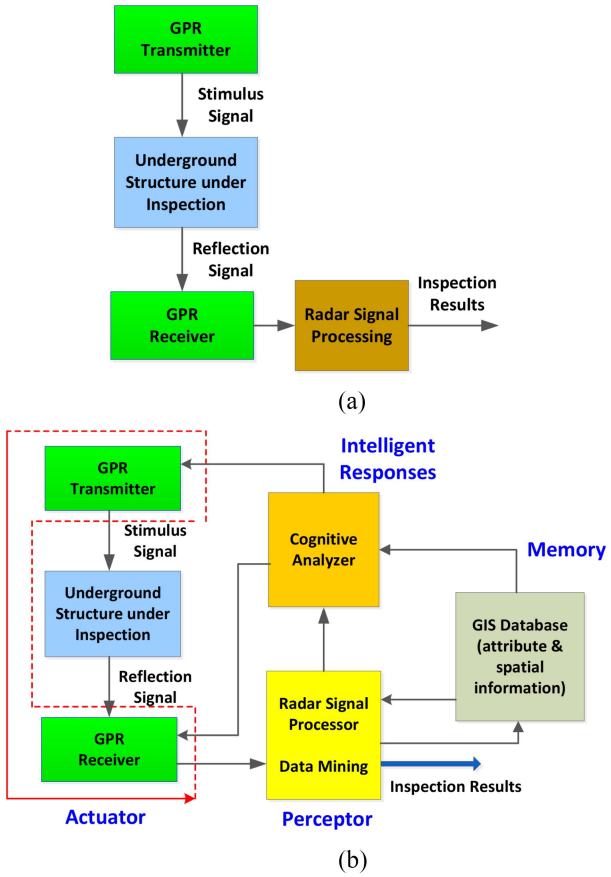

Fig. 3. Operational flows of traditional and cognitive GPRs. (a) Traditional GPR. (b) Cognitive GPR.

efficiency of subsurface sensing. Fig. 3(a) shows the conventional mode of GPR operation where an expert sets the operational parameters into an optimal configuration based on previous experience with similar past situations. This is, an iterative time-consuming process. The conventional mode is not suitable for continuous long-time operations, especially in a complex environment inaccessible to humans.

The concept of cognitive GPR was proposed in [27] where intelligence was expected to be generated on the fly to adaptively adjust the operational parameters based on data analysis and feedback control. As shown in Fig. 3(b), the cognitive

TABLE I
NOMENCLATURE

Deep Q-network	
$s_t \equiv \langle \kappa_t, \Psi_t \rangle$	The state of AC-GPR agent at time step t κ_t is the observation about scene Ψ_t is the state of the agent such as position and battery energy status
s_{t+1}	The succeeding state s_t of the agent after time step t
$a_t \equiv \langle \xi_t, \vec{v}_t, \vec{p}_t \rangle$	The action AC-GPR agent will take, which involve moving direction ($\xi_t \in [0, 2\pi]$), velocity, and the configuration (e.g., frequency and orientation) of GPR
$\epsilon \in [0, 1]$	The probability that a random action is taken
$r_t \in \Re$	Reward at time step t
$r_t^i \in \Re$	Rényi entropy-based reward
$r_t^m \in \Re$	Shannon entropy-based reward
$\gamma \in \Re$	Discount factor, of the target Q-value
D and L	Replay memory and its dimension
$Q(\theta)$	Evaluation Q-network that's updated every episode, where θ is the network parameter
$\hat{Q}(\hat{\theta})$	Target Q-network which is updated every predefined episodes, where $\hat{\theta}$ is the network parameter
π^*	Optimum policy
Ground Penetration Radar Signal	
τ	Time-interval of the reflection signal from the object
$\langle \Delta\tau, \Delta\varpi \rangle$	Dimension of the region-of-interest in a B-Scan
ε	Dielectric constant
$z_i(\tau)$, $z_j(\varpi)$	Power normalization wrt two way time and scan axis
$E_\alpha(\tau)$, $E_\alpha(\varpi)$	Entropy values wrt two way time and scan axis
$\Im 1^*$, $\Im 2^*$	Optimum Otsu thresholds
$Z(\tau)$	Reflection signal
E_{min}, E_{max}	Minimum and maximum Rényi entropy

GPR consists of an adaptive GPR transceiver, a perceptor module, a memory module, and a cognitive analyzer. The operation of the cognitive GPR follows a perception-action cycle: first, the GPR transceiver collects the reflected wave data about subsurface objects and sends them to the preceptor. Then, the preceptor processes and analyzes the data to extract signature patterns and format a perception of subsurface conditions. The memory module has a GIS database containing urban subsurface condition attributes and spatial locations. The cognitive analyzer carries out machine learning based on both the processing results from the perceptor and the prior knowledge about GPR measurement from the memory module to produce intelligent responses, locally or at the edge [28] for the control of radar transceiver reconfigurations.

Although the concept of cognitive GPR in [27] pointed to promising direction in the system architecture development, no unambiguous definition or approach for a cognitive GPR was provided to build an adaptive and smart GPR that generates intelligence to adaptively adjust its operational parameters in an uncertain and dynamic sensing environment.

B. MDP Formulation of Cognitive GPR Operation

The cognitive control of the positioning and operational parameters of a GPR can be formulated as a sequential decision-making problem which can be further modeled as a finite-horizon MDP with finite state and action spaces.

Without loss of generality, we consider a discrete-time system in which time is divided into slots of unit length ΔT such that each slot t corresponds to the time duration $[(t-1)\cdot\Delta T, t\cdot\Delta T)$. The complete notation used in this article is given in Table I.

The MDP model is described as follows.

Authorized licensed use limited to: UTC Library. Downloaded on October 01,2021 at 17:00:56 UTC from IEEE Xplore. Restrictions apply.

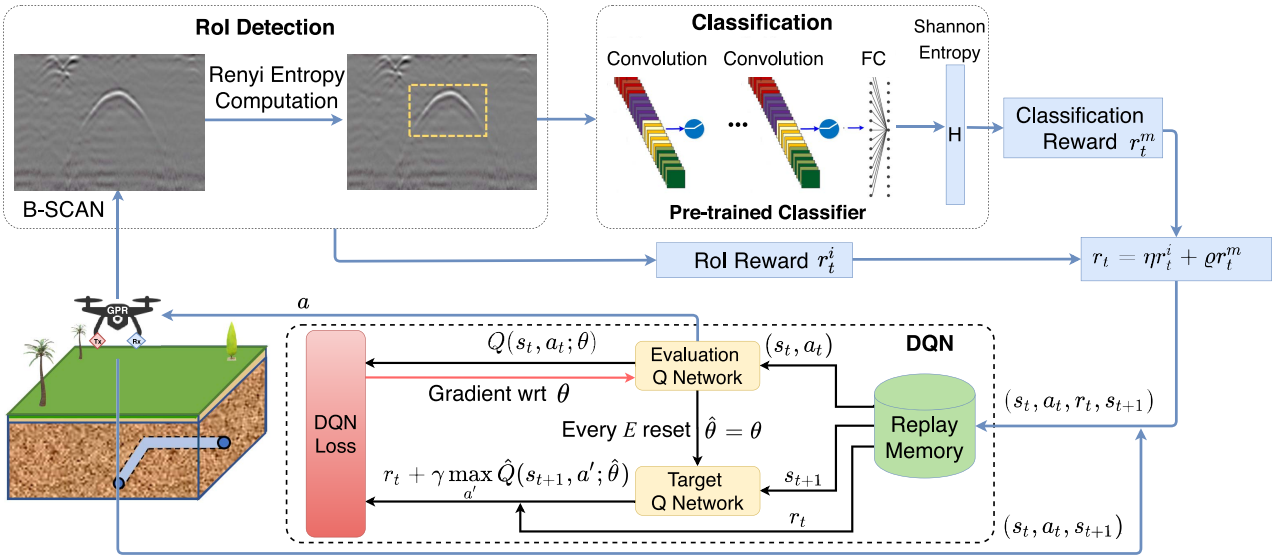

Fig. 4. Iterative operational process of the proposed DRL-enabled AC-GPR.

1) $\mathcal{S}$: A set of environment and system operational states. Let $s_t = (\kappa_t, \Psi_t) \in \mathcal{S}$ denote the state of the GPR sensing system and the environment in each discrete time slot t. κ_t is the newly updated observation about the environment, in the form of captured B-Scan image. Ψ_t is the operating state vector of the GPR, such as the remaining battery energy of the mobile GPR platform and the agent's position $X_t \in \mathbb{C}$ (a complex number), i.e., $X_t = x_t + jy_t$, representing the GPR location with coordinates (x_t, y_t).
2) $\mathcal{A}$: A set of actions of the GPR. Let $a_t = (\xi_t, \vec{v}_t, \vec{p}_t) \in \mathcal{A}$ denote the action vector to be performed at time step t where $\vec{p}_t$ is the operational parameter values of the GPR; $(\xi_t, \vec{v}_t)$ denote the moving direction and velocity of the GPR platform, respectively. Thus, the position of the GPR at time step t can be derived as $X_t = X_{t-1} + \vec{v}_t \cdot \Delta T \cdot e^{j\xi_t}$.
3) $P_t(s, a, s') = Pr(s_{t+1} = s' | s_t = s, a_t = a)$: The probability of transition from state s to state s' under action a.
4) K: Horizon over which the GPR will act.

In the proposed research, the core problem of the MDP is to find a "policy" for the GPR: a function π that specifies the action $a_t = \pi(s_t)$ that the GPR will choose in state s_t to maximize its accumulative knowledge about the subsurface object over horizon K : $\mathbb{E}[\sum_{t=0}^{T} \gamma^t r_t(s_t, a_t)]$ where $\mathbb{E}[.]$ is the expectation taken over $s_{t+1} \sim P(s_{t+1}|s_t, a_t)$ and $\gamma (0 \leq \gamma \leq 1)$ is the discount factor of the reward at different time steps. Due to the extreme curse of dimensionality in the state space $\mathcal{S}$ and the immense challenge of identifying transition probability $P(s_{t+1}|s_t, a_t)$, it is impractical to use exact methods such as linear programming and dynamic programming to solve the MDP problem.

To overcome this challenge, we will investigate a DRL framework where an AC-GPR agent is reinforced to learn a policy. As a computational methodology for automated decision-making of intelligent agents in uncertain environments, DRL has progressed tremendously in the past decade [24], [29]. DRL is concerned with how a decision-making agent ought to take actions from a given state of an environment so as to maximize some notion of cumulative reward. The full potential of DRL requires the agent to directly interact with the environment to attain a flow of real-world experience, as shown in Fig. 2.

C. Architecture of the Proposed AC-GPR

In this section, we present an overview of the proposed AC-GPR architecture, as shown in Fig. 4. The architecture has an iterative operational process involving environment observation, reward identification, DQN-based policy learning, and action execution. The observations (B-Scan images) from the AC-GPR agent are feed into a RoI detection module where a RoI, for example, an image area including a hyperbolic signature resulting from a subsurface rebar, is identified and extracted through the image segmentation technique using Rényi entropy and Otsu method [30], [31]. The pretrained classifier receives the RoI image as input for classification. The classification probability output is used to characterize the classification confidence. The output results from the RoI module and classification module are used to form the reward for the AC-GPR agent, which will be described in Section IV-A.

The DQN module takes a tuple of state, action, reward, and future state as experience, and guides the AC-GPR agent to learn an optimum policy that maximizes the future discounted reward. The algorithm of the DQN module will be described in Section IV-B. As one of value-based DRL methods, DQN is considered because it provides a better sample efficiency and more stable performance compared with policy gradient methods that have drawback of high variance in estimating the gradient.

IV. Proposed DRL Approach

The cognitive analyzer in Fig. 3(b) is a critical component of the proposed AC-GPR. It produces intelligence to direct the GPR movement and its operational configurations

Authorized licensed use limited to: UTC Library. Downloaded on October 01,2021 at 17:00:56 UTC from IEEE Xplore. Restrictions apply.

based on the collected GPR data and prior knowledge about GPR measurement. This section presents a DRL approach to the implementation of the cognitive analyzer with a novel rewarding mechanism.

A. Reward Function

The AC-GPR agent is rewarded through the outcome of RoI detection and object classification while interacting with the environment. The reward function $r : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is derived by combining two types of rewards that are computed based on Rényi entropy and Shannon entropy, respectively, that effectively characterize the AC-GPR agent's newly acquired subsurface knowledge about the subsurface object from the sensory data.

The rationale behind this combination of the two subrewards is that the AC-GPR agent would receive reward r_t^i when it identifies an RoI in the B-scan image and receive reward r_t^m when it recognizes some object properties, such as the diameter and material of a subsurface pipeline, through GPR data classification.

Thus, the overall reward function is

$$r_t(s_t, a_t) = \eta r_t^i + \varrho r_t^m \tag{1}$$

where r_t^i denotes the RoI detection reward; r_t^m denotes object recognition reward; and η and ϱ denote the weight coefficients whose values are determined based on the relative importance of the RoI detection reward and the object recognition reward.

In this section, we briefly present the concept of RoI detection reward in Section IV-A-1, and then describe subsurface object classification reward in Section IV-A-2.

1) Reward Based on RoI Detection: Prior to the RoI detection, a B-Scan image first goes through the following preprocessing steps: signal denoising by removing the DC component (arithmetic mean) from each trace of the GPR image, time-zero correction which adjusts all traces to a common time-zero position where the first break of air-wave is observed, and signal enhancement/amplification [3]. The preprocessed B-Scan image is input to the RoI module in Fig. 4 to identify a peculiar area of the image, such as a hyperbola signature, through computing Rényi entropy and Otsu threshold, generating an extracted RoI image. Rényi entropy is preferred because of its high level of accuracy on signal processing tasks compared to Tsallis [32]–[34]. Rényi entropy has been considered in vast domains such as, structure health monitoring clutter rejection for intrawall diagnostics [32], tracking electroencephalographic signals changes [33], and cardiac autonomic neuropathy in diabetic patients [34].

In this work, Rényi entropy is calculated to recognize the singular region on a B-scan image. In particular, a high Rényi entropy value demonstrates high level of information similarity while a low Rényi entropy value features high level of information peculiarity [30]. Let $Z(\tau)$ denote the collected GPR reflection signal which can be depicted as

$$Z(\tau) = D(\tau) + \zeta(\tau) \tag{2}$$

where $D(\tau)$ represents the reflection signal from the object of interest, and $\zeta(\tau)$ models remaining interference and noise

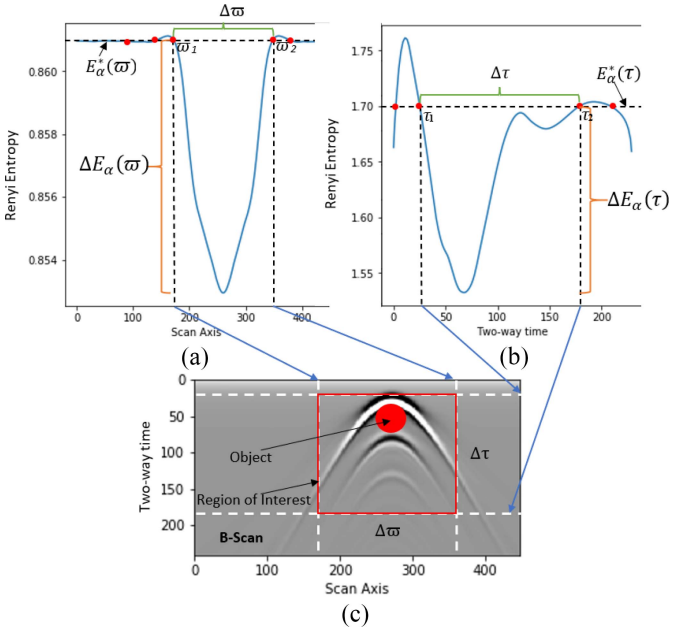

Fig. 5. Plots (A) and (B) shows Rényi entropy versus scan axis and Rényi entropy versus two-way time, respectively. Plot (C) shows the dimension of the Region of Interest (RoI) as the red box.

upon preprocessing. In calculation, power normalization is first performed with the summation of the power of the same time index data points on different traces, which can be expressed as

$$z_i(\tau) = \frac{||Z_i(\tau)||^2}{\sum_{i=1}^{I} ||Z_i(\tau)||^2} \tag{3}$$

where $z_i(\tau)$ is the normalized signal, i is the trace index, I is the total number of traces included; τ is the time index of pulse data on each reflection trace waveform. Upon power normalization, to assess data singularity over the wave travel time axis (that is, y-axis) of the B-scan, a generalized Rényi entropy is calculated as

$$E_\alpha(\tau) = \frac{1}{1-\alpha} \log_e \sum_{i=1}^{I} [z_i(\tau)]^\alpha \tag{4}$$

where $E_\alpha(\tau)$ is the entropy quantification, and α denotes the entropy order. Equation (4) is equivalent to the basic Shannon entropy limiting value as $\alpha \rightarrow 1$.

Similarly, Rényi entropy calculation is applied to the scanning traces axis (that is, x-axis)

$$z_j(\varpi) = \frac{||z_j(\varpi)||^2}{\sum_{j=1}^{J} ||z_j(\varpi)||^2} \tag{5}$$

where $z_j(\varpi)$ is the normalized signal, j is the time index of pulse and J is the total number of time indexes; $\varpi = \delta t \cdot \vec{v}$ is the displacement along the trace axis of the pulse data. Then, as shown in Fig. 5(a) the Rényi entropy to assess data singularity over the scanning position along the x-axis of the B-scan is

$$E_\alpha(\varpi) = \frac{1}{1-\alpha} \log_e \sum_{j=1}^{J} [z_j(\varpi)]^\alpha. \tag{6}$$

Authorized licensed use limited to: UTC Library. Downloaded on October 01,2021 at 17:00:56 UTC from IEEE Xplore. Restrictions apply.

Fig. 5(a) and (b) show the entropy plots with respect to the trace axis and two-way travel time axis of a B-Scan image, respectively.

With two selected entropy thresholds $\Im 1$ and $\Im 2$, the B-scan image can be segmented into three classes of nonoverlapping regions: 1) singular region; 2) stationary background region; and 3) the transition region in-between. The singular region entropy values are lower than threshold $\Im 1$, the stationary background region entropy values are higher than $\Im 2$. While for the transitioning region, its entropy values are between these two thresholds.

The optimum thresholds $\Im 1^*$ and $\Im 2^*$ are determined through the Otsu method [31], a classic image segmentation technique for extracting an object from its background. Specifically, by initializing both $\Im 1$ and $\Im 2$ at zero, the Otsu method performs statistical analysis to identify appropriate thresholds so as to segment image into different regions based on the criteria: the intensity values variances of the same region is minimized while the variances of different regions are maximized. In this work, when applying the Otsu method, the entropy is chosen as the intensity value [30].

Fig. 5(a) and (b) shows the calculated upper bound Otsu thresholds: $\Im 2^* = 0.8601$ over the scan axis and $\Im 2^* = 1.70$ over the travel time axis, respectively. By identifying Rényi entropy values that intersect with the thresholds, boundary trace values (i.e., $\varpi_1 = 182$ and $\varpi_2 = 363$) and travel time values (i.e., $\tau_1 = 26$, $\tau_2 = 178$) can be determined with low entropy values contained in the resulting intervals. Then, these four boundary values are superimposed on Fig. 5(c), demarcating the RoI, as shown by the red box. As an example, Fig. 5(c) shows the resulting RoI including a section of hyperbola where the highest value marked by the red ball is the position of the object, such as a rebar or a pipe.

From the computed Rényi's probability distributions $E_\alpha(\tau)$ and $E_\alpha(\varpi)$, coupled with the optimum Otsu thresholds $\Im 2_t^* = E_\alpha^*(\tau)$ and $\Im 2_s^* = E_\alpha^*(\varpi)$, the reward for detecting the RoI is computed as

$$r_t^i = a(\Delta E_\alpha(\tau) \cdot \Delta E_\alpha(\varpi)) + b(\Delta\tau \cdot \Delta\varpi) \quad (7)$$

$$\Delta E_\alpha(\tau) = E_\alpha^*(\tau) - \min\{E_\alpha(\tau)\} \quad (8)$$

$$\Delta E_\alpha(\varpi) = E_\alpha^*(\varpi) - \min\{E_\alpha(\varpi)\} \quad (9)$$

$$\Delta\tau = \tau_2 - \tau_1 \quad (10)$$

$$\%\Delta\varpi = \varpi_2 - \varpi_1 \quad (11)$$

where $\Delta E_\alpha(\tau)$ and $\Delta E_\alpha(\varpi)$ denote the Rényi entropy variation with respect to the travel time axis and the scanning position axis, respectively; $\Delta\tau$ denotes the related two-way travel time interval and $\Delta\varpi$ the related scanning position interval, indicating the dimension of the detected RoI; and a and b are weight coefficients whose values are determined based on the relative importance of the Rényi entropy variation and the RoI dimension; As higher data singularity corresponds to lower Rényi entropy, higher $\Delta E_\alpha(\tau)$ and $\Delta E_\alpha(\varpi)$ indicate higher chance of detecting the subsurface object [30]. The proposed reward function in (7) combines the Rényi entropy change with the RoI dimension, mitigating the false positive detection resulting from outliers.

2) Reward Based on Subsurface Object Classification: Subsurface objects can be recognized through different GPR data classification tasks, for example, determining the material type, burial depth, and diameter depending on specific applications. In GPR data processing and analysis, as shown in the top part of Fig. 4, the RoI identified through the Rényi entropy computation is passed to a pretrained convolutional neural network (CNN) classifier. The use of CNN is motivated by the fact that CNNs outperform other artificial neural networks on conventional computer vision tasks, such as object detection [35], facial expression recognition [36], and medical imaging segmentation [37], through feature learning. Let $P = \{p(1), p(2), \ldots, P(N)\}$ denote the classification probability output from the classifier where $p(n)$ is the class probability that the processed B-scan belongs to class n, and N is the total number of classes. The possible classes depend on the specific classification task. For example, if the classification task is to determine the material type of the subsurface object, the possible classes could be different material types, namely concrete, metallic, polyvinyl chloride (PVC), etc. As entropy is a measure of uncertainty [38], [39], in this work, Shannon entropy is considered to quantify the confidence in the classification. The Shannon entropy of the classification probability distribution P can be computed as

$$r_t^m = -\sum_{n=1}^{N} p(n)\log(p(n)). \quad (12)$$

It is inferred from (12) that a balanced classification probability distribution results in high entropy indicating high uncertainty and low classification confidence while a skewed classification probability distribution has low entropy indicating low uncertainty and high classification confidence.

B. DRL Algorithm

In RL, an agent learns to better perform tasks by learning from its experiences interacting with the environment. Our proposed DRL method is based on the DQN algorithm taking four inputs (s, a, r, s'), i.e., state observation, action, reward, next state observation, via epsilon-greedy strategy.

As shown in Fig. 4 the DQN algorithm has three main components: 1) the evaluation Q-network ($Q(s, a; \theta)$); 2) the target Q-network ($\hat{Q}(s, a; \hat{\theta})$); and 3) the replay memory. The evaluation Q-network and the target Q-network have the same network structure but different weights and biases. However, the evaluation Q-network is updated instantly for every episode, whereas the target Q-network is updated periodically after every H episodes as shown in Table III, by replacing the values of the target Q-network with the evaluation Q-network values. The target network is updated only infrequently in order to mitigate the risk of nonstationarity of the target values in the loss function in (15) caused by the feedback loops between the target and estimated Q-values. The generated Q-values, through a replay memory with random batch size of B as shown in Table III, are used to compute the DQN loss in (15).

Authorized licensed use limited to: UTC Library. Downloaded on October 01,2021 at 17:00:56 UTC from IEEE Xplore. Restrictions apply.

The AC-GPR is inclined to execute the action with the highest Q-value derived from each episode. The Q-value corresponding to the pair of state and action represents an expected discounted accumulated future reward.

As discussed in Section IV-A, the AC-GPR agent received reward $r_t = r(s_t, a_t)$ at each time instance t. The accumulated reward is defined as $R_t = \sum_{t=0}^{T} \gamma^t r_t$ with a discounting factor $\gamma \in [0, 1]$. The action-value function under action policy π is defined as $Q^{\pi}(s_t) = \mathbb{E}[R_t | s_t, \pi]$, and the optimal policy is determined by estimating the action-value function Q^{π}. The estimation can be obtained by Bellman equation recursively [40]. An episode is terminal on one of the following two conditions: 1) when AC-GPR detects a subsurface object or 2) when AC-GPR observes a B-Scan with a predefined number of traces (A-Scans).

To improve the convergence of DQN, the agent's experience at each time step t is stored at the replay memory, where a batch B of experiences are randomly sampled, as shown in Table III. This process is called experience replay that provides diverse and decorrelated training data and solves the issue of correlated inputs/output [41]. Let e_t represent the agent's experience at time t which is defined as

$$e_t = (s_t, a_t, r_t, s_{t+1}). \tag{13}$$

This tuple contains the state s_t, the action a_t taken at state s_t, the reward r_t given to the agent at time t as a result of the previous state-action pair (s_t, a_t), and the next state s_{t+1}. The replay memory is set to a finite size limit L, and therefore, it will only store the latest L experiences.

The DQN utilizes a CNN as a nonlinear approximation to the optimal action-value function $Q(\theta) \rightarrow Q^*$. All parameters of the network are denoted as θ, and the parameters are estimated iteratively by minimizing the temporal difference error as

$$\hat{\theta} = \arg\min_{\theta} \mathbb{E}\Bigg[\Bigg(r_t + \gamma \max_{a_{t+1}} \hat{Q}\Big(s_{t+1}, a_{t+1}; \hat{\theta}\Big) - Q(s_t, a_t; \theta)\Bigg)^2\Bigg]. \tag{14}$$

The optimization is converted into a regression problem in DQN by regarding temporal difference error as loss

$$L(\theta) = \left[\underbrace{r_t + \gamma \max_{a_{t+1}} \hat{Q}\Big(s_{t+1}, a_{t+1}; \hat{\theta}\Big)}_{\text{target y}} - Q(s_t, a_t; \theta)\right]^2. \tag{15}$$

In the training process, y is regarded as the target of the regression function. The loss function is used to train the neural network to adjust parameters θ.

After training, parameters θ^* are obtained, and the algorithm stops after a finite number of steps M with the optimal policy $V^{\pi_M} = V^*$. With approximation, the optimum policy maximises the future discounted reward as

$$\forall s \;\; \pi^*(s_t) = \arg\max_{a} \sum_{s_{t+1}} P_{s_t, a_t}(s_{t+1}) V^*(s_{t+1}) \tag{16}$$

where $P_{s_t, a_t}(s_{t+1})$ is the state transition probability as discussed in Section III-B, and V^* the optimum value function with convergence $\pi_{t+1}(s_{t+1}) = \pi_t(s_t)$. Note that the computation of both $P_{s_t, a_t}(s_{t+1})$ and the sum in (16) is avoided due to the action-value function approximation through $Q(\theta)$.

Algorithm 1: Cognitive Subsurface Object Detection Algorithm Based on Deep Q-Network

1 Initialize replay memory D to capacity L
2 Initialize Q with random parameter θ
3 Initialize $\hat{Q}$ with weights $\hat{\theta} = \theta$
4 **for** *episode = 1, M* **do**
5 Initialize state s_1
6 **for** *t=1, T* **do**
7 Select a random action a_t with probability ϵ;
8 Otherwise select $a_t = arg\max_a Q(s_t, a; \theta)$;
9 Execute action a_t, capture B-Scan image, and compute reward $r_t = \eta r_t^i + \varrho r_t^m$;
10 Set $s_{t+1} = (x_{t+1}, y_{t+1})$,
11 Store transition (s_t, a_t, r_t, s_{t+1}) in D;
12 Sample random batch (s_j, a_j, r_j, s_{j+1}) from D;
13 Calculate target y_j

$$y_j = \begin{cases} r_j & \text{if terminal } s_{j+1} \\ r_j + \gamma \max_{a'} \hat{Q}\Big(s_{j+1}, a'; \hat{\theta}\Big) & \text{else;} \end{cases}$$

  Perform a gradient descent step in (15) with respect to network parameters θ;
14 Every H steps reset $\hat{Q} = Q$, i.e., set $\hat{\theta} = \theta$.
15 **end**
16 **end**

After learning the policy, the agent needs to perform action selection and execution. The agent faces the well-known exploration-exploitation dilemma of whether to exploit the current knowledge by following the learned policy or to continue to explore the uncertain environment to acquire more knowledge. To resolve the dilemma, the agent adopts an ϵ-greedy-based policy for selecting actions where $\epsilon \in [0, 1]$ denotes the exploration probability. The configuration that $\epsilon = 0$ would result in the pure greedy policy that always selects the action corresponding to the highest Q-value. The pure greedy method may cause the policy learning to get stuck at a local optima. In contrast, the configuration that $\epsilon = 1$ would result in the pure random policy that selects an action randomly without considering the Q-value.

1) Data Set: A nonzero value of $\epsilon \in (0, 1)$ would allow the agent to select random actions with probability of ϵ regardless of the actual Q-value. A decaying ϵ would allow the agent to select random actions with decreasing probabilities as the learning goes on. A decaying ϵ may help to stabilize the learned policy which converges to an optimal one.

As described in Algorithm 1, at first (lines 1–3), the network parameters are initialized randomly. To enhance the learning stability, the target network is introduced which has the same structure as the evaluation network. Then, the exploration process is conducted, and the action is derived from the DQN. The DQN employs the RoI and classifier modules to update the

Authorized licensed use limited to: UTC Library. Downloaded on October 01,2021 at 17:00:56 UTC from IEEE Xplore. Restrictions apply.

proposed reward function (line 9), by combining the RoI detection reward and object classification reward, which intuitively measures the amount of the acquired information about the subsurface object from the observed B-Scan image. Next (lines 11–14), the experiences are stored into the replay memory. Then, the minibatch method is used to randomly collect examples from the replay memory. The weights and biases of the network are updated by training the DQN according to the loss function (15). The training process will terminate once it reaches a predefined number of episodes. During each episode, the AC-GPR agent stops performing actions after a predefined number of time steps, or could terminate the episode early if it detected the subsurface object. The computational complexity of AC-GPR algorithm is expressed as $\mathcal{O}(MT)$, where M denotes the total number of episodes and T the number of time steps.

V. Performance Evaluation and Discussion

In this section, we systematically evaluate the performance of the proposed AC-GPR by using a GPR simulator called GprMax [5], [6] designed for modeling GPR operations. B-scan images are generated by GprMax on the fly in mimicing the GPR data acquisition in the real environment.

A. Experiment Settings

1) GprMax Simulator: The GprMax simulator used in this study solves 2-D Maxwell equations using the finite-difference time-domain (FDTD) method [42]. The simulation in this work considers small diameter pipes or rebars made of three types of materials, namely, concrete, metallic and PVC, as subsurface objects. GprMax characterizes the impact of common pipe materials on resulting GPR data based on the dielectric constant also called relative permittivity which indicates how easily a material can become polarized. Relative permittivity, defined as the ratio of the permittivity of a substance to the permittivity of space of vacuum, is expressed as $\varepsilon = (C/C_o)$, where C denotes the capacitance of the material as the dielectric capacitor, $C_o = (\varepsilon_o A/d)$ represents the capacitance using vacuum as the dielectric, ε_o the permittivity of free space (8.85 × 10-12 F/m, i.e., Farad per meter), A the area of the sample cross section area, and d the thickness of the sample. The dielectric constant for PVC, concrete and metal are 4.0, 4.94, and infinity, respectively.

GprMax uses a mixing model for modeling radio propagation in soil [43]. The soil composition involves sand fraction 0.3, clay fraction 0.1, bulk density 2 g/cm^3, sand particle density of 2.66 g/cm^3, and a volumetric water fraction range of 0.001–0.25.

The GPR data set contains B-Scan images from the GprMax simulator, which was used to train the classifier, as shown in Fig. 4. Fig. 6 shows some example B-Scan images which are corresponding to concrete, metallic and PVC objects. Some images have sharp, dim or no hyperbola due to different factors including, but not limited to, object material, burial depth, soil dielectric properties, GPR antenna configuration and orientation.

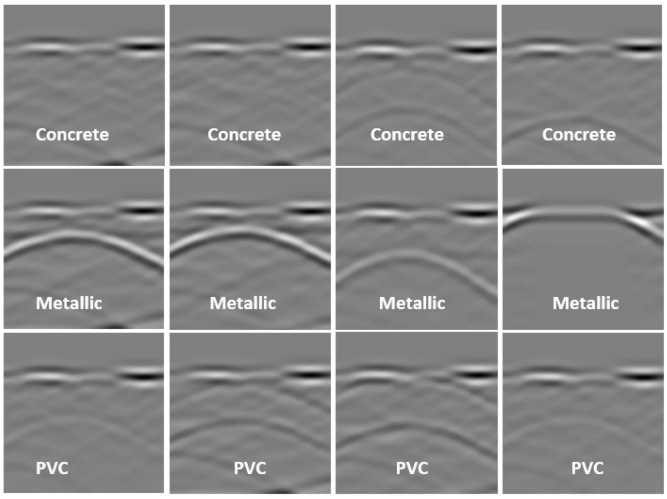

Fig. 6. Sample B-Scans of the data set used to the train the classifier. The data set includes 1440 concrete, 1440 metallic, and 1440 PVC B-Scans.

In this work, we generated a total of 40 320 images with each material type having 13 440 B-Scan images, derived from varying soil dielectric properties and object diameter. All the 40 320 images are unique, each exhibiting different level of contrast, shape and size on hyperbola signature. The B-Scans from concrete objects exhibit weak or no hyperbola at all due to the fact that the signals are attenuated as they propagate through the soil and through the object hence weakening the signal reflection. The PVC objects have a slightly higher dielectric constant than concrete objects; accordingly, they produce B-Scans with slightly high hyperbola contrast. B-Scans from metallic objects have strong and high resolution hyperbola contrast due to the high relative permittivity of metals which allows for strong reflected signals.

2) RoI Detection: The RoI detection module in Fig. 4 receives a preprocessed B-Scan image and identifies a possible RoI with the method detailed in Section IV-A. The RoI is resized into a 64 ×64 grayscale images which will be feed into the classification module, as illustrated in Fig. 5.

3) Object Classification: The pretrained classifier in Fig. 4 is configured with two hidden convolutional layers. The first layer has 32 filters, kernel size 3 applied with stride 2, while the second one has 64 filters, kernel size 3 with stride 2. All two layers are followed by a LeakyReLU nonlinearity. This is followed by a fully connected layer with 256 units, and three-class output layer and softmax, learning rate of 10^{-3} and batch size of 64. Based on the classification outcome, i.e., the probability values of different classes, Shannon entropy is calculated. During training a high class prediction probability means low uncertainty hence a high confidence of a particular class. Table II summarizes the hyperparameters of the CNN-based object classifier.

4) DQN: Simulations are conducted on a core i7 computer with four cores, 2.2-GHz Intel Xeon CPU, and 16-GB RAM. The training process is run with Python 3.6 and tensorflow 1.10.0. The size of the replay memory is 5×10^4, and the sample mini batch is $B = 32$. During the training process, the GPR agent interacts with the environment and receives tuples

Authorized licensed use limited to: UTC Library. Downloaded on October 01,2021 at 17:00:56 UTC from IEEE Xplore. Restrictions apply.

TABLE II
ARCHITECTURE PARAMETERS OF THE CNN-BASED OBJECT CLASSIFIER, AND THE EVALUATION AND TARGET NETWORKS OF DQN

Layer (type)	CNN Object Classifier	DQN Evaluation Network & Target Network
Conv2D	Filter: 32 Kernel size: 3 Stride: 2 Padding: Same Activation: LeakyReLU	Filter: 256 Kernel size: 3 Activation: ReLU
MaxPooling	Pool size: 2	Pool size: 2
Dropout	Rate: 0.3	Rate: 0.5
Conv2D	Filter: 64 Kernel size: 3 Stride: 2 Padding: Same Activation: LeakyReLU	Filter: 256 Kernel size: 3 Activation: ReLU
MaxPooling	Pool size: 2	Pool size: 2
Flatten	Output shape: 1024	Output shape: 50176
Dense	Classes: 3	Classes: $\mathcal{A}$

TABLE III
DEEP Q-NETWORK SETTINGS

Hyperparameter	Value	Description
Learning rate (α)	0.01	The learning rate used in ADAM to update the DQN
Discount factor (γ)	0.99	The discount factor used in the reinforcement update
Epsilon (ϵ)	$[0, 1]$	Exploration probability
Replay memory size (L)	5×10^4	Number of the most recent experiences stored in the replay memory
Mini batch size (B)	32	Number of experiences used for training
Epsilon decay	0.99975	Exploration probability decay factor
Episodes (M)	15000	Number of Episodes
Frequency of target network update (H)	5	Number of time steps before the target network is updated

of state, action, reward and next state. A total of 5×10^4 such tuples are stored in the replay memory as experiences which is then sampled and used during the learning. The agent starts by exploring the environment to build knowledge about transitions and action rewards. Then, through decaying the exploration probability ϵ, the agent gradually exploits the gathered information to detect subsurface objects. Tables II and III summarize the architecture parameters and network hyperparameters of DQN, respectively.

B. Performance Results

1) Likelihood of Successful Object Detection Versus Noisy B-Scan Data: In this article, we evaluate the performance of the proposed AC-GPR under different levels of clutter noise caused by clutter from heterogeneous soil.

Based on the model of radio propagation proposed by Peplinski [43], heterogeneous soils are modeled by considering sand fraction, clay fraction, bulk density, sand particles density, and the range for volumetric water fraction. The noise levels are modeled through adjusting both sand and clay fraction from 0.1 to 0.9. The clutter noise level increases as the fraction value increases. To make the clutter noise levels more distinct, different types of soil surfaces including smooth, rough, water, and grass surfaces, are added to the fractal-box (a box that houses Peplinski heterogeneous soil).

Interpreting noisy B-scan data caused by clutter noise is challenging, and sometimes it is impossible to extract some

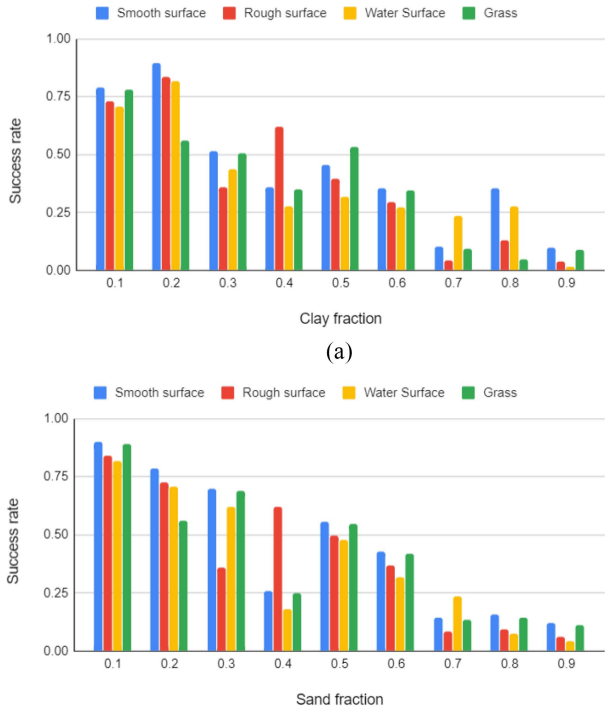

Fig. 7. AC-GPR performance with noisy B-Scan images collected from soils with varying clay and sand fractions. (a) Clay fraction analysis. (b) Sand fraction analysis.

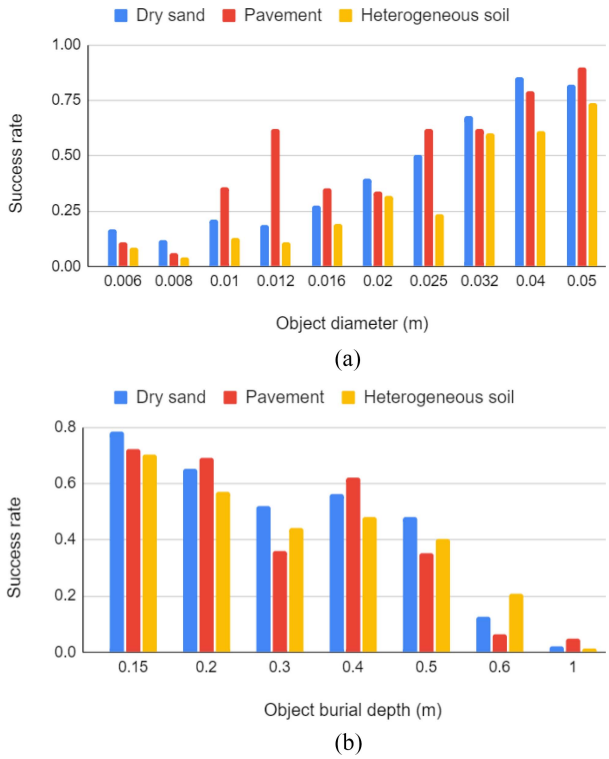

Fig. 8. AC-GPR performance with varying (a) object diameter and (b) object burial depth.

Authorized licensed use limited to: UTC Library. Downloaded on October 01,2021 at 17:00:56 UTC from IEEE Xplore. Restrictions apply.

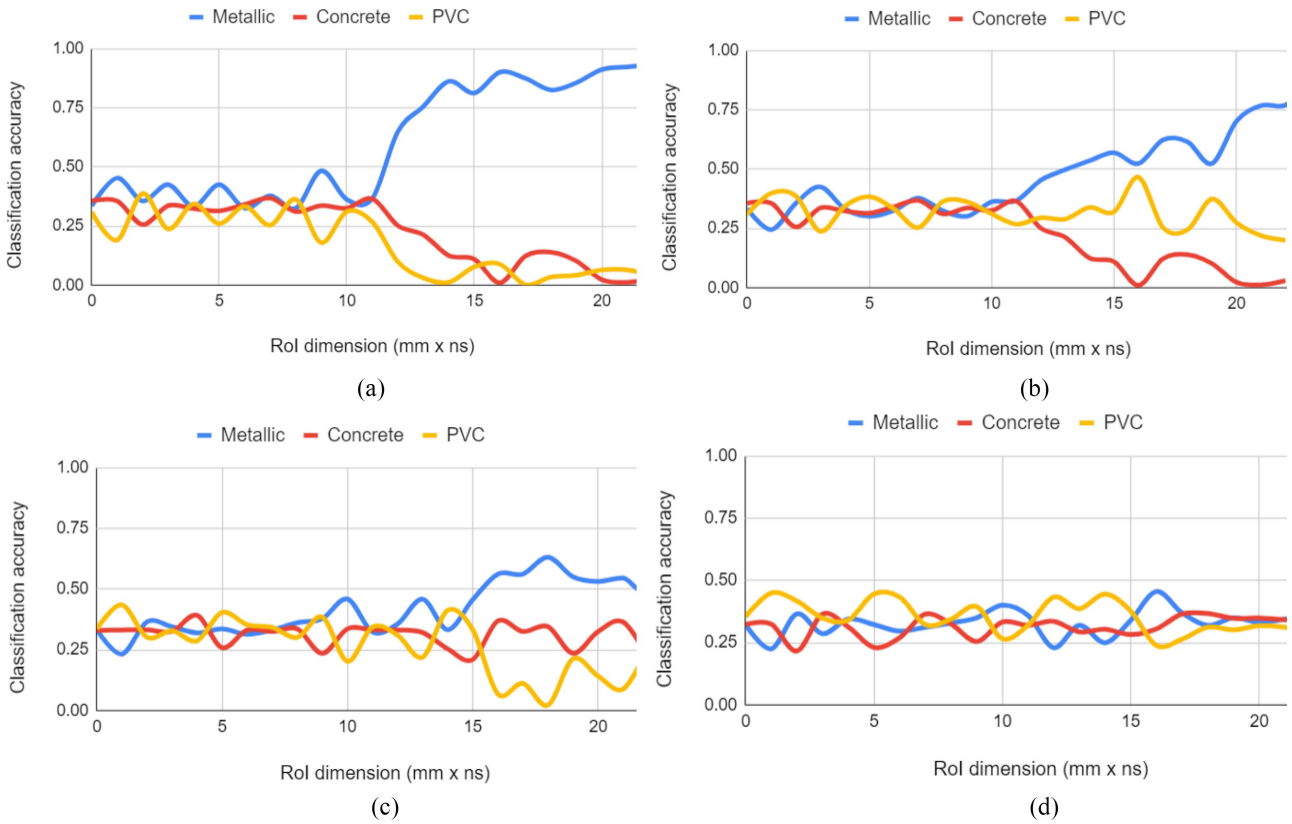

Fig. 9. Classification accuracy versus RoI dimension for metallic, concrete and PVC objects buried in four different media: (a) dry sand, (b) wet sand, (c) pavement, and (d) heterogeneous soil.

knowledge about the subsurface object from the data. In this work, the likelihood of successful object detection and recognition is used to evaluate the performance of the proposed AC-GPR on this aspect. Specifically, a performance metric termed success rate is defined as $(w/(w+l))$ where w denotes the number of episodes with successful object detection, and l the number of episodes with failed detection. Successful object detection and recognition are characterized by high confidence results of object detection and material type classification, respectively, from a B-Scan image. Fig. 7 shows that the resulting success rate decreases along with the increase of the level of the clutter noise caused by the increased clay or sand fraction.

2) Likelihood of Successful Object Detection Versus Object Diameters and Burial Depths: To simulate real-world scenarios, the performance of the proposed AC-GPR was also tested through modeling various object diameters and burial depths in dry sand, pavement and heterogeneous soil. Fig. 8(a) shows that as the diameter of the object increases the success rate increases. This is because with a larger object diameter more signals are reflected back to the GPR receiver, hence generating B-Scans with higher resolution hyperbolas. As shown in Fig. 8(b), the success rate decreases as the burial depth of the object increases. This is because the deeper the object is buried the more attenuation the signals incur as they propagate through the soil, resulting in weaker reflection and fainter hyperbolas that makes it more challenging for the AC-GPR to detect and recognize the subsurface object.

3) Classification Accuracy Versus RoI Dimension: As GPR data interpretation is affected by the size of the detected RoI, the impact of RoI dimension on the classification accuracy of the classifier was also evaluated. Fig. 9 shows the classification accuracy versus RoI dimension for metallic, concrete, and PVC objects buried in four different media: 1) dry sand; 2) wet sand; 3) pavement; and 4) heterogeneous soil. As shown in Fig. 9(a), the classification for metallic objects in dry sand has the highest accuracy. The level of accuracy increases as the RoI dimension increases. Fig. 9(b) shows a slight decline in classification accuracy in wet sand with we compared with the result of dry sand. That is, because the more water content in the wet sand caused more signal attenuation. Fig. 9(c) and Fig. 9(d) shows the low classification accuracy in Pavement and heterogeneous soil, respectively. This is, because the severe signal attenuation in the pavement and heterogeneous soil resulted in weak signature of object in the RoI, such as a hyperbola. Even though the RoI dimension increases, the classifier fails to produce a high level of confidence about the classification of the object material.

C. Convergence Analysis

1) Comparison Between Reward Functions: In order to evaluate the proposed reward function, we study the impact of different reward function variations on performance convergence. In the evaluation, three types of rewards were considered, that is, the reward only from RoI detection, the reward only from object classification, and the combined

Authorized licensed use limited to: UTC Library. Downloaded on October 01,2021 at 17:00:56 UTC from IEEE Xplore. Restrictions apply.

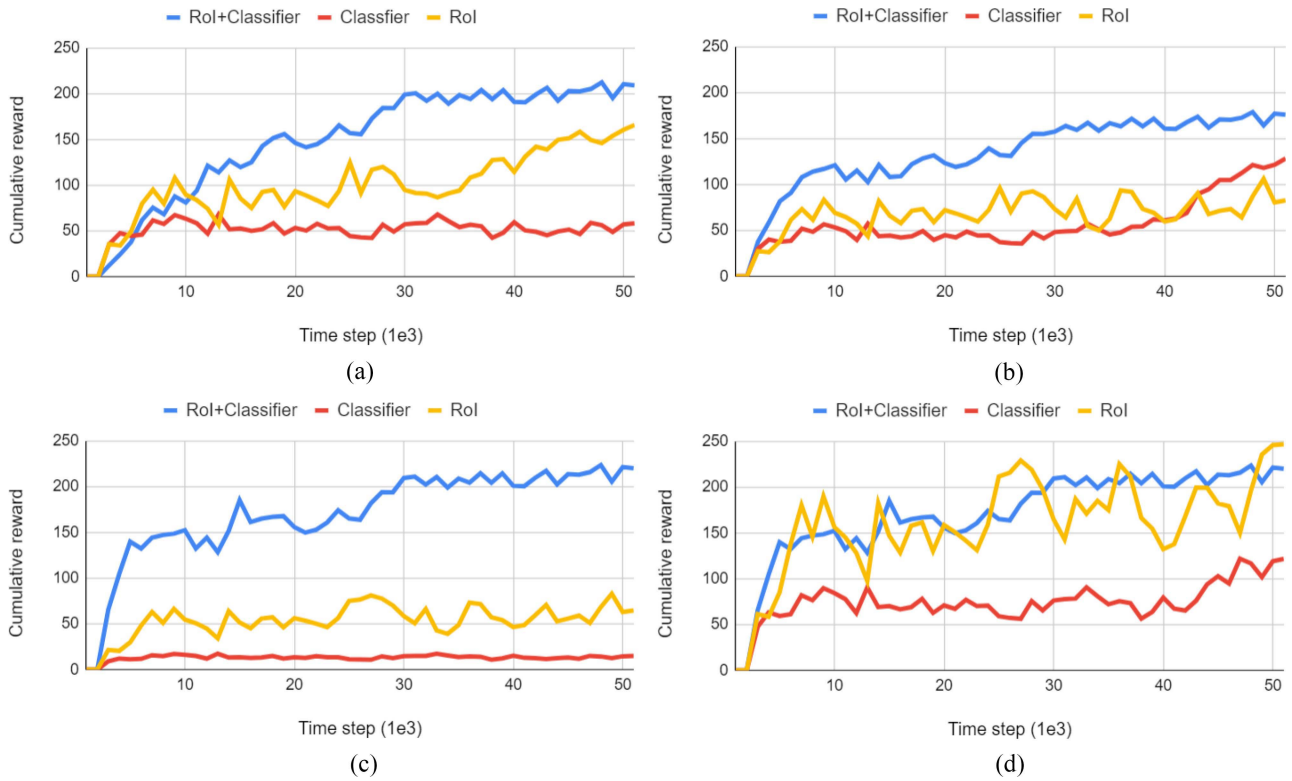

Fig. 10. Cumulative reward versus time steps with different ϵ-greedy policies and types of rewards. (a) AC-GPR decay ($\epsilon = 1 - 0.001$). (b) AC-GPR ($\epsilon = 0.001$). (c) AC-GPR ($\epsilon = 1$). (d) Q-Learning.

TABLE IV
COMPARISON OF THE TIME STEPS NEEDED FOR ALGORITHM 1 TO REACH CONVERGENCE WITH DIFFERENT ϵ-GREEDY POLICIES AND TYPES OF REWARDS

Model	RoI Detection	Object Classification	RoI Detection + Object Classification
AC-GPR (ϵ decay)	1.53×10^3	1.34×10^3	1.28×10^3
Q-Learning	3.56×10^3	4.07×10^3	3.39×10^3
AC-GPR ($\epsilon = 0.001$)	4.23×10^3	4.19×10^3	4.04×10^3
AC-GPR ($\epsilon = 1$)	7.27×10^3	7.15×10^3	7.12×10^3

rewards from both RoI detection and object classification. The rewards were computed from a heterogeneous soil setup containing varying clay and sand fractions (0.1−0.9) with an object having a diameter of 0.05 m at a burial depth of 0.3 m. For comparison, different AC-GPRs with different ϵ-greedy policies were investigated. In addition, a Q-Learning-based method was also evaluated by considering a fairly small state space. Fig. 10 shows the cumulative reward versus time step in different cases. It is observed that the AC-GPR using the proposed combined rewards outperforms the systems using other types of rewards.

2) Time Steps to Reach Convergence: The time needed for the proposed AC-GPR to reach converged performance was also evaluated. The time steps to reach convergence for different AC-GPR implementations adopting different ϵ-greedy policies are displayed in Table IV. It is shown that the proposed AC-GPR configured with decaying-epsilon-greedy policy (ϵ decay) outperforms the rest. Additionally, the proposed AC-GPR using the combined reward demonstrates better convergence performance.

VI. CONCLUSION

In this article, an autonomous cognitive GPR (AC-GPR) based on DRL was proposed. A novel reward function was developed such that the AC-GPR agent is rewarded from both RoI detection and object classification. With the proposed reward function a DQN-based model was developed to enable the AC-GPR to learn to take optimal actions that maximizes the long-term discounted reward, hence detecting and identifying subsurface objects from its experiences of interacting with the environment. The proposed AC-GPR was evaluated by adapting the GPR operation simulator, GprMax, and simulating real-world environment and GPR operations. Simulation results show the proposed AC-GPR has superior performance over other GPR systems in terms of object detection success rate, object classification accuracy, and convergence.

It is worth noting that the proposed approach for AC-GPR is suitable for the detection of a single subsurface object with simple structural configurations, such as a pipe or a rebar, and under relatively homogeneous environment. However, in real-world environment, detection of subsurface objects involves multiple challenges including inherent uncertainties and complexities of the environment for EM wave propagation and GPR data acquisition, scarcity of ground-truth GPR data set for model training, and structural heterogeneity of different

Authorized licensed use limited to: UTC Library. Downloaded on October 01,2021 at 17:00:56 UTC from IEEE Xplore. Restrictions apply.

subsurface objects. As our future work, advanced machine learning algorithms and models for an AC-GPR that incorporate effective GPR signal processing methods and GPR data processing approaches will be developed for detecting and modeling objects in complicated environments.

REFERENCES

[1] H. M. Jol, *Ground Penetrating Radar Theory and Applications*. London, U.K: Elsevier, 2009.

[2] *WaveSense*. Accessed: Jun. 25, 2019. [Online]. Available: https://wavesense.io

[3] A. Turk, A. Hocaoglu, and A. Vertiy, *Subsurface Sensing*. Hoboken, NJ, USA: Wiley, 2011.

[4] W. B. Powell, *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. Hoboken, NJ, USA: Wiley, 2007.

[5] *Gprmax*. Accessed: Nov. 10, 2020. [Online]. Available: https://www.gprmax.com/about.shtml

[6] C. Warren, A. Giannopoulos, and I. Giannakis, "gprMax: Open source software to simulate electromagnetic wave propagation for ground penetrating radar," *Comput. Phys. Commun.*, vol. 209, pp. 163–170, Dec. 2016.

[7] P. P. Ray, "Internet of Robotic Things: Concept, technologies, and challenges," *IEEE Access*, vol. 4, pp. 9489–9500, 2016.

[8] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "Safe, multi-agent, reinforcement learning for autonomous driving," 2016. [Online]. Available: http://arxiv.org/abs/1610.03295

[9] A. E. Sallab, M. Abdou, E. Perot, and S. Yogamani, "Deep reinforcement learning framework for autonomous driving," *Electron. Imag.*, vol. 2017, no. 19, pp. 70–76, 2017.

[10] M. Cornick, J. Koechling, B. Stanley, and B. Zhang, "Localizing ground penetrating radar: A step toward robust autonomous ground vehicle localization," *J. Field Robot.*, vol. 33, no. 1, pp. 82–102, 2016.

[11] R. M. Williams, L. E. Ray, and J. Lever, "An autonomous robotic platform for ground penetrating radar surveys," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2012, pp. 3174–3177.

[12] V. Kafedziski, S. Pecov, and D. Tanevski, "Detection and classification of land mines from ground penetrating radar data using faster R-CNN," in *Proc. 26th Telecommun. Forum (TELFOR)*, 2018, pp. 1–4.

[13] W. Rice, M. M. Omwenga, D. Wu, and Y. Liang, "Enhanced underground object detection with conditional adversarial networks," in *Proc. ISSAT Int. Conf. Data Sci. Intell. Syst.*, 2019.

[14] A. Foessel-Bunting, D. Apostolopoulos, and W. L. Whittaker, "Radar sensor for an autonomous Antarctic explorer," in *Mobile Robots XIII and Intelligent Transportation Systems*, vol. 3525, H. M. Choset *et al.*, Eds. Bellingham, WA, USA: Int. Soc. Opt. Photon. SPIE, 1999, pp. 117–124. [Online]. Available: https://doi.org/10.1117/12.335690

[15] T. Le, S. Gibb, N. Pham, H. M. La, L. Falk, and T. Berendsen, "Autonomous robotic system using non-destructive evaluation methods for bridge deck inspection," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 3672–3677.

[16] T. Sugimoto and M. Gouko, "Acquisition of hovering by actual UAV using reinforcement learning," in *Proc. 3rd Int. Conf. Inf. Sci. Control Eng. (ICISCE)*, pp. 148–152, Jul. 2016.

[17] W. Xia, H. Li, and B. Li, "A control strategy of autonomous vehicles based on deep reinforcement learning," in *Proc. 9th Int. Symp. Comput. Intell. Design (ISCID)*, 2016, pp. 198–201.

[18] A. Murad, F. A. Kraemer, K. Bach, and G. Taylor, "Autonomous management of energy-harvesting IoT nodes using deep reinforcement learning," 2019. [Online]. Available: http://arxiv.org/abs/1905.04181

[19] J. Xin, H. Zhao, D. Liu, and M. Li, "Application of deep reinforcement learning in mobile robot path planning," in *Proc. Chin. Autom. Congr. (CAC)*, Oct. 2017, pp. 7112–7116.

[20] H. Kretzschmar, M. Spies, C. Sprunk, and W. Burgard, "Socially compliant mobile robot navigation via inverse reinforcement learning," *Int. J. Robot. Res.*, vol. 35, no. 11, pp. 1289–1307, 2016. [Online]. Available: https://doi.org/10.1177/0278364915619772

[21] Y. Zhu *et al.*, "Target-driven visual navigation in indoor scenes using deep reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 3357–3364.

[22] H. Li, Q. Zhang, and D. Zhao, "Deep reinforcement learning-based automatic exploration for navigation in unknown environment," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 6, pp. 2064–2076, Jun. 2020.

[23] D. Zhao, Y. Chen, and L. Lv, "Deep reinforcement learning with visual attention for vehicle classification," *IEEE Trans. Cogn. Develop. Syst.*, vol. 9, no. 4, pp. 356–367, Dec. 2017.

[24] X. Han, H. Liu, F. Sun, and X. Zhang, "Active object detection with multistep action prediction using deep Q-network," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3723–3731, Jun. 2019.

[25] E. Lin, Q. Chen, and X. Qi, "Deep reinforcement learning for imbalanced classification," 2019. [Online]. Available: http://arxiv.org/abs/1901.01379

[26] D. Liu and T. Jiang, "Deep reinforcement learning for surgical gesture segmentation and classification," 2018. [Online]. Available: http://arxiv.org/abs/1806.08089

[27] S. Haykin, "Cognitive radar: A way of the future," *IEEE Signal Process. Mag.*, vol. 23, no. 1, pp. 30–40, Jan. 2006.

[28] D. Wu, M. M. Omwenga, Y. Liang, L. Yang, D. Huston, and T. Xia, "A fog computing framework for cognitive portable ground penetrating radars," in *Proc. IEEE ICC*, May 2019, pp. 1–6.

[29] M. A. Abd-Elmagid, A. Ferdowsi, H. S. Dhillon, and W. Saad, "Deep reinforcement learning for minimizing age-of-information in uav-assisted networks," 2019. [Online]. Available: http://arxiv.org/abs/1905.02993

[30] Y. Zhang, P. Candra, G. Wang, and T. Xia, "2-D entropy and short-time fourier transform to leverage GPR data analysis efficiency," *IEEE Trans. Instrum. Meas.*, vol. 64, no. 1, pp. 103–111, Jan. 2015.

[31] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 9, no. 1, pp. 62–66, Jan. 1979.

[32] R. Solimene and A. D'Alterio, "Entropy-based clutter rejection for intrawall diagnostics," *Int. J. Geophys.*, vol. 2012, May 2012.

[33] J. Lerga, N. Saulig, and V. Mozetič, "Algorithm based on the short-term Rényi entropy and if estimation for noisy eeg signals analysis," *Comput. Biol. Med.*, vol. 80, pp. 1–13, Jan. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0010482516302888

[34] D. J. Cornforth, M. P. Tarvainen, and H. F. Jelinek, "Using Rényi entropy to detect early cardiac autonomic neuropathy," in *Proc. 35th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, 2013, pp. 5562–5565.

[35] S. Gidaris and N. Komodakis, "Object detection via a multi-region and semantic segmentation-aware CNN model," in *Proc. IEEE Int. Conf. Comput. Vision (ICCV)*, Dec. 2015, pp. 1134–1142.

[36] Y. Li, J. Zeng, S. Shan, and X. Chen, "Occlusion aware facial expression recognition using CNN with attention mechanism," *IEEE Trans. Image Process.*, vol. 28, no. 5, pp. 2439–2450, May 2019.

[37] B. Kayalibay, G. Jensen, and P. van der Smagt, "CNN-based segmentation of medical imaging data," 2017. [Online]. Available: http://arxiv.org/abs/1701.03056

[38] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, 1948.

[39] A. Namdari and Z. Li, "A review of entropy measures for uncertainty quantification of stochastic processes," *Adv. Mech. Eng.*, vol. 11, no. 6, pp. 1–14, 2019.

[40] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[41] S. Gu, E. Holly, T. Lillicrap, and S. Levine, "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 3389–3396.

[42] K. S. Kunz and R. J. Luebbers, *The Finite Difference Time Domain Method for Electromagnetics*. Boca Raton, FL, USA: CRC Press, 1993.

[43] N. R. Peplinski, F. T. Ulaby, and M. C. Dobson, "Dielectric properties of soils in the 0.3-1.3 GHz range," *IEEE Trans. Geosci. Remote Sens.*, vol. 33, no. 3, pp. 803–807, May 1995.

Maxwell M. Omwenga (Graduate Student Member, IEEE) received the B.S. and M.S. degrees in computer science from Makerere University, Kampala, Uganda, in 2006 and 2012, respectively. He is currently pursuing the Ph.D. degree with the University of Tennessee at Chattanooga, Chattanooga, TX, USA.

From 2012 to 2016, he worked with Bugema University, Kampala, as a Lecturer. His research interests include reinforcement learning, signal processing, and edge computing.

Dr. Omwenga is an Award Winner of the Technology Symposium 2019 and a member of *Sigma Xi*.

Authorized licensed use limited to: UTC Library. Downloaded on October 01,2021 at 17:00:56 UTC from IEEE Xplore. Restrictions apply.

Dalei Wu (Member, IEEE) received the B.S. and M.Eng. degrees in electrical engineering from Shandong University, Jinan, China, in 2001 and 2004, respectively, and the Ph.D. degree in computer engineering from the University of Nebraska–Lincoln, Lincoln, NE, USA, in 2010.

Since August 2014, he has been with the Department of Computer Science and Engineering, University of Tennessee at Chattanooga, Chattanooga, TN, USA, where he is currently an Associate Professor. From October 2011 to June 2014, he was a Postdoctoral Research Associate with the Mechatronics Research Laboratory, Massachusetts Institute of Technology, Cambridge, MA, USA. His current research interests include data-driven intelligent systems, cyber–physical systems, real-time decision making, and complex dynamic system modeling, and optimization.

Yu Liang received the B.E. degree in computer science and technology from Tsinghua University, Beijing, China, in 1990, the M.S. degree in computer science from Beijing University of Technology, Beijing, in 1995, the first Ph.D. degree in computer science from the Institute of Computing Technology of Chinese Academy of Sciences, Beijing, China, in 1998, and the second Ph.D. degree in applied mathematics from the University of Ulster at Coleraine, Northern Ireland, U.K., in 2005.

He is currently a Professor with the Department of Computer Science and Engineering, University of Tennessee at Chattanooga, Chattanooga, TN, USA. His current research interests focus on modeling and simulation, machine learning and artificial intelligence, big-data and cloud computing, parallel computing, numerical linear algebra, and computational science.

Li Yang received the M.S. and Ph.D. degrees in computer science from Florida International University, Miami, FL, USA, in 2003 and 2005, respectively.

She is currently a Guerry Professor and the Director of Information Security Center, University of Tennessee at Chattanooga (UTC), Chattanooga, TN, USA, and the National Center of Academic Excellence in Information Assurance/Cyber Defense, UTC. Her research interests include network and information security, mobile security, big data analytics, massive data mining, bioinformatics, cybersecurity education, and engineering techniques for complex software system design.

Dryver Huston received the B.S. degree from the University of Pennsylvania, Philadelphia, PA, USA, in 1980, and the M.A. and Ph.D. degrees from Princeton University, Princeton, NJ, USA, in 1983 and 1986, respectively.

Since 1987, he has been a Faculty Member with the University of Vermont, Burlington, VT, USA. He has authored the book, structural sensing, health monitoring, and performance evaluation.

Tian Xia (Senior Member, IEEE) received the B.E. degree in electrical engineering from Huazhong University of Science and Technology, Wuhan, China, in 1994, the M.S. degree from Nanjing University of Posts and Telecommunications, Nanjing, China, in 2000, and the Ph.D. degree in electrical and computer engineering from the University of Rhode Island, South Kingstown, RI, USA, in 2003.

He is currently a Professor with the Department of Electrical and Biomedical Engineering, University of Vermont, Burlington, VT, USA. He has over 150 publication in scientific conferences and journals. His research interests focus on microelectronic circuit and system design and test, sensor circuit design and applications, and signal processing.

Authorized licensed use limited to: UTC Library. Downloaded on October 01,2021 at 17:00:56 UTC from IEEE Xplore. Restrictions apply.